

### **REMARKS**

These remarks are responsive to the Office Action, dated March 6, 2008. Currently, claims 1-20 are pending with claims 1, 8 and 13 being independent. Claims 1-3, 5, 7-8, and 13 have been amended to expedite prosecution of this application to allowance and to overcome Examiner's objections. Claims 16-20 have been added. Support for these amendments is found in the Applicants' specification at least on page 8, lines 3-13; page 9, line 21 to page 10, line 27; page 14, line 24 to page 18, line 19.

#### **Interview Summary**

Pursuant to 37 C.F.R. 1.133 and MPEP 713.04, Applicant submits the following Statement of Substance of the Interview.

Applicants would like to thank the Examiner for the opportunity to discuss the above application during a telephonic interview on June 25, 2008. During the interview, Examiner Michael D. Pham (along with another Examiner) and Applicant's representative Boris A. Matvenko were present. The following is a summary of the conducted interview.

- (1) no exhibits were discussed or shown at the interview;
- (2) Claims 1-15 were discussed;
- (3) U.S. Patent Publication No. 2004/0049513 to Yakir et al. (hereinafter, "Yakir"), U.S. Patent No. 5,991,753 to Wilde (hereinafter, "Wilde"), U.S. Patent Publication No. 2002/0055972 to Weinman, Jr. (hereinafter, "Weinman"), and U.S. Patent No. 5,564,037 to Lam (hereinafter, "Lam") references were discussed;
- (4) The Examiner and Applicant's representative have discussed Applicants' claims in relation to the Yakir, Wilde, Weinman, and Lam references. Applicants pointed out that Yakir

fails to disclose all elements of the present invention, in that Yakir requires presence of the originating and destination servers in order to operate, whereas the present invention has no such requirement. Further, Yakir tracks location of the file and when an entire file is desired, Yakir transfers entire contents of the file and fails to maintain, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver and then using the lists, initiating recovery of files in the set of files on the destination fileserver, as recited in claim 1. Applicants further pointed out that Wilde fails to cure the deficiencies of Yakir. Wilde performs file migration using bit files and generates a list of files eligible for migration as well as deals with special handling of files. However, Wilde also fails to disclose the maintaining step of the present invention. Weinman is concerned with maintaining a specific number of file copies in the network, but is not concerned with their specific location, as such, it also fails to disclose the maintaining step of claim 1. Lam simply discusses use of stub files and as such does not cure the deficiencies of either Yakir, Wilde, or Weinman, or their combination. Applicants further pointed out numerous advantages of the present invention, including the present invention allowing disaster recovery when originating fileserver is not operational, faster recovery process, ability replicate metadata across multiple volumes, thus, no requiring HSM-assisted disaster recovery, and that the present invention does not require manual access and reload of files and does not use automatic reloading of cached data. The Examiner alleged that the combination of the above references still discloses the subject matter of the claims and suggested amending the maintaining step to further describe how the recovery of the files is performed by the present invention.

(5) The Examiner and Applicants did not reach any agreements.

(6) No other matters were discussed during the interview.

35 U.S.C. 103(a)

In the March 6, 2008 Office Action, the Examiner rejected claims 1-4, and 8 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Publication No. 2004/0049513 to Yakir (hereinafter, "Yakir") in view of U.S. Patent No. 5,991,753 to Lam (hereinafter, "Lam"). In the Office Action, the Examiner stated that Yakir discloses all elements of claim 1 except that it does not "explicitly disclose 'repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination server.'" (Office Action, page 4). The Examiner stated that Wilde discloses this element. Applicants respectfully disagree and traverse this rejection.

Amended claim 1 recites, *inter alia*, receiving, at a destination fileserver, a set of stub files associated with the set of files, maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, said initiating further comprises using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file, and, replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files, replacing the stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files.

As understood by Applicants, Yakir relates to techniques for moving a stub file from an originating storage location to a destination storage location without recalling the migrated data corresponding to the stub files. (Yakir, Abstract). Further, Yakir discloses an advanced Historical Storage Management (“HSM”) based storage system that allows shares of data to be migrated from an originating server to a destination server. As Applicants pointed out in their previous response and during June 25, 2008 interview, one of the major drawbacks of Yakir is that it requires that both originating and destination file servers are present in order to migrate files and is incapable of performing any data migration when originating file server is inactive. (emphasis supplied). This is in contrast to the present invention that provides disaster recovery operation when originating file server is non-operational, failed and/or non-existent. Yakir clearly cannot perform such task. Further, Yakir cannot replicate metadata across multiple volumes and, as such, it cannot perform HSM-assisted disaster recovery.

Yakir moves stub files from an originating storage location to a destination storage location without recalling migrated data corresponding to the stub file. (Yakir, para. [0009]). Yakir’s originating and destination storage locations can be on the same storage unit and assigned to the same or different file servers. (Yakir, para. [0009]). Yakir’s storage management system (“SMS”) includes information that relates to location of files that were migrated (or re-migrated) and recalled. (Yakir, para. [0023]). The information can also include information related to storage policies, rules for storage environment, information related to various monitored storage units, information related to files stored in the storage environment, file location information that includes information used to find location of migrated data. (Yakir, para. [0023]). File location information can be replicated to databases on servers. (Yakir, para. [0023]).

Yakir includes file location information using which Yakir's server management system ("SMS") finds the file. In contrast to the present invention, Yakir fails to maintain a list of repository nodes that contain a replica of the file. Instead, Yakir stores "file location information or portions thereof" in various storage locations. (Yakir, para. [0023]). The data locator information is stored in the stub file. (Yakir, para. [0071]). However, Yakir fails to disclose, teach or suggest, maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, as recited in claim 1. Further, as stated above, Yakir is incapable of operating when an originating fileserver no longer exists. As such, Yakir fails to disclose, teach or suggest initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, as recited in claim 1. Instead, Yakir simply uses file location information stored in the stub file to obtain the full content of the file. However, if the Yakir's file location information points to a location that no longer exists, Yakir cannot obtain the full content of the file.

Additionally, Yakir also fails to disclose replacing each stub file with a full content of the file associated with the stub file, wherein the replacing includes receiving a client request for a specified file in the set of files, replacing the stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files, as recited in claim 1. Yakir simply replaces a requested stub file (using file location information stored in the stub files) with its full content, but does not continue to replace other non-requested files, which is contrary to the recitation of claim 1. Hence, Yakir does not disclose, teach or suggest all elements of claim 1.

Wilde does not cure the deficiencies of Yakir. As understood by Applicants and as pointed out to the Examiner during June 25, 2008 Interview, Wilde discloses a file management system for implementing special handling of files for migration, compression, encryption and logging access to files. (Wilde, Abstract). Wilde distinguishes between a resident file, which is a file with all of its contents stored on a local disk, and a non-resident file, which is a file that has been migrated. (Wilde, Col. 7, lines 35-39). During migration, a list of all the files in the file system that are eligible for migration along with migration attributes is generated. The files are ranked by weighing their age and size factors and then sorted according to the ranking. The resulting list is called a candidate list. (Wilde, Col. 8, lines 11-21). However, similar to Yakir, Wilde fails to disclose maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, as recited in claim 1. Instead, Wilde simply creates a ranked list of files for migration, but does not maintain a list of files in the system and the list of repository nodes, where replica of each file is stored, contrary to the recitation of claim 1. (emphasis supplied).

Also, Wilde fails to disclose initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, as recited in claim 1. Instead, Wilde is concerned with migration techniques associated with resident files (stored on the local disk) and non-resident files (files that have been migrated), as well as, creating a ranked list of files that are subject to migration. This is clearly different than present invention's recovering of files based on the information about where replicas of files are stored.

Additionally, Wilde also fails to disclose allowing access to a full content of a file associated with the stub file, and, replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files, replacing the stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files, as recited in claim 1. Wilde's system does not allow replacement of files based on priority and then background replacement of remaining files. Wilde creates a list of files eligible for migration and then migrates them, which makes the process of migration very slow, contrary to the present invention.

As previously pointed out the Examiner, Weinman also does not cure the deficiencies of either Yakir, Wilde, or their combination. Wienman, as many other conventional HSM systems, is extremely slow in its file recovery efforts. The present invention allows faster recovery and access to files. Typically, in conventional systems, after a disaster or failure of a storage site, to allow access to files to users, an administrator would have to manually access and reload files that were previously stored at the failed storage site to a backup storage site. This process takes a lot of time and users cannot immediately access their files. This is contrary to the present invention that allows users through the use of stub files (4 Kb in size) to quickly access most needed files and while users are accessing the necessary files, the remaining files are being transferred to the backup site.

As understood by Applicants, Weinman discloses a data dispersion method for reducing a risk of losing a file by replicating it across geographically-distant locations. (Weinman, paras. [0014]-[0016]). Upon creation of an object, Weinman mirrors the object to n-1 additional mirror sites in the network. The number of copies may change upon creation of additional copies of

disasters occurring in the additional sites. (Weinman, para. [0015]). Weinman also determines whether too few or too many copies have been created and either adds or deletes copies of files. (Weinman, para. [0015]). As such, Weinman is only concerned with maintaining a specific number of copies of files on the network, but is not concerned with maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, as recited in the amended claim 1. Weinman does not maintain any lists of nodes that have a replica of a file. Weinman also fails to disclose initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, said initiating further comprises using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file, and, replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files, replacing the stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files, as recited in claim 1.

Lam also does not cure the deficiencies of either one of the above references or their various combinations. As understood by Applicants, Lam relates to a real time data migration in a networked computer system using a sparse file to represent a migrated file, where the sparse file consumes a minimum amount of physical space, but has all the attributes of the actual file. (Lam, Abstract). However, Lam fails to disclose maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the



set of files stored at the destination fileserver, initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, said initiating further comprises using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file, and, replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files, replacing the stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files, as recited in claim 1.

One having ordinary skill in the art would not look to either Wilde, Weinman, and/or Lam to solve the deficiencies of Yakir. Even though all four references are in the same field of endeavor, all four references are deficient as they fail to disclose the maintaining and initiating steps of the present invention. Additionally, neither one of the references seeks to improve the speed of file recovery.

Further, as previously pointed out in Applicants' October 31, 2007 response, the above references and their respective technologies significantly differ from each other and provide no basis as to why they would be combined. Specifically, Yakir relates to moving of stub files and storage management; Wilde relates to creation of migration candidates lists; Weinman relates to maintenance of a specific number of file copies in a network; and, Lam relates to use of stub. Neither Yakir, nor Wilde, nor Weinman, nor Lam provide any suggestion or motivation as to why they should be combined. Thus, the only suggestion and/or motivation to combine Yakir, Wilde, Weinman, and/or Lam may be found in the present application, however, such

suggestion/motivation cannot be relied upon in combining the reference. Hence, it is improper to combine Yakir, Wilde, Weinman, and/or Lam without some disclosed motivation other than the present application. See, MPEP 2143.01:

“There are three possible sources for a motivation to combine references: the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art.” *In re Rouffet*, 149 F.3d 1350, 1357, 47 USPQ2d 1453, 1457-58 (Fed. Cir. 1998) (The combination of the references taught every element of the claimed invention, however without a motivation to combine, a rejection based on a prima facie case of obvious was held improper.).

Even if one were to combine Yakir, Wilde, Weinman, and/or Lam, which would be improper, the present invention is not realized. Yakir relates to techniques for moving a stub file from an originating storage location to a destination storage location, where such techniques use data-locator information stored in the stub file. Wilde relates to creation and use of ranking candidate migration lists. Weinman's system creates multiple copies of actual files in various geographically-distant locations. Lam relates to use of stub files. The various combinations of Yakir, Wilde, Weinman, and/or Lam fail to disclose, *inter alia*, maintaining, at the destination fileserver, a list of repository nodes that contain a replica of each file in the set of files, and a list of files in the set of files stored at the destination fileserver, initiating recovery of files in the set of files on the destination fileserver, wherein based on the list of files and the list of repository nodes stored at the destination fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes, said initiating further comprises using a stub file in the set of stub files, allowing access to a full content of a file associated with the stub file, and, replacing each stub file with a full content of the file associated with the stub file, wherein said replacing includes receiving a client request for a specified file in the set of files, replacing the

stub file associated with the specified file with a full content of the specified file, and, replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files, as recited in claim 1.

Applicants further incorporate herein by reference in their entireties Applicants' arguments submitted in their responses, dated March 23, 2007 and October 31, 2007, as well as, arguments presented during the June 25, 2008 Examiner's Interview.

Based on the above, Applicants respectfully submit that claim 1 is not rendered obvious by various combinations of Yakir, Wilde, Weinman, and/or Lam. Thus, the rejection of claim 1 is respectfully traversed. The Examiner is requested to reconsider and withdraw his rejection of claim 1.

Claims 2- 15 are patentable over various combinations of Yakir, Wilde, Weinman, and/or Lam for at least the reasons stated above with regard to claim 1. As such, the rejections of claims 2-15 are respectfully traversed. The Examiner is requested to reconsider and withdraw his rejections of claims 2-15.

New claims 16-20 are patentable over the various combinations of Yakir, Wilde, Weinman, and/or Lam for at least the reasons stated above with regard to claim 1.

**CONCLUSION**

No new matter has been added. The claims currently presented are proper and definite. Allowance is accordingly in order and respectfully requested. However, should the Examiner deem that further clarification of the record is in order, we invite a telephone call to the Applicants' undersigned attorney to expedite further processing of the application to allowance.

Applicants believe that no additional fees are due with the filing of this Amendment. However, if any additional fees are required or if any funds are due, the USPTO is authorized to charge or credit Deposit Account Number: 50-0311, Customer Number: 35437, Reference Number: 25452-015.

Respectfully submitted,



Date: July 7, 2008

Boris A. Matvenko, Reg. No. 48,165  
MINTZ LEVIN COHN FERRIS  
GLOVSKY & POPEO, P.C.  
Chrysler Center  
666 Third Avenue, 24<sup>th</sup> Floor  
New York, NY 10017  
Tel: (212) 935-3000  
Fax: (212) 983-3115